



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/531,260	04/13/2005	James Scott Tarbell	PTB-4942-5	4670
23117 7590 05/10/2010 NIXON & VANDERHYE, PC 901 NORTH GLEBE ROAD, 11TH FLOOR ARLINGTON, VA 22203				
EXAMINER				
CHOL YUK TING				
ART UNIT		PAPER NUMBER		
2164				
MAIL DATE		DELIVERY MODE		
05/10/2010		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/531,260  
Filing Date: April 13, 2005  
Appellant(s): TARBELL ET AL.

NIXON & VANDERHYE P.C  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 03/01/2010 appealing from the Office action mailed 07/27/2009.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**GROUND OF REJECTION NOT ON REVIEW**

The following grounds of rejection have not been withdrawn by the examiner, but they are not under review on appeal because they have not been presented for review in the appellant's brief. Claim 67 stand rejected under 35 U.S.C 103 (a) as allegedly being unpatentable by Bills, Tanaka and further in view of Cloud (US Patent 6,253,369 B1).

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

US2002/0152195 A1	Bills et al	10-2004
US Patent 6,665,735	Tanaka et al	12-2003
US2003/0217031 A1	Owen et al	11-2003
US Patent 6,829,768	Suzuki et al	12-2004
US Patent 6,253,369	Cloud et al	06-2001

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

**Claims 46-56, 62-66, 68-78 and 84-91 are rejected under 35 U.S.C. 103(a) as being unpatentable by Bills (US PA Pub 2002/0152195 A1) and in view of Tanaka (US Patent 6,665,735).**

Referring to claim 46, Bills discloses disclose a computer implemented method in a database journal changes to system objects in an operating system (*See para. [0005] and para. [0014], journaling objects changes, e.g. journaling objects when objects are created or deleted in an operating system*) with a processor (*See Fig. 1, item 21*), the method including:

generating copies of system objects for journaling (*See para. [0058], journaling an object when a system creates or modifies an existing object*).

Bills does not explicitly disclose executing a dummy function in place of a system function when the system function is called and executing the system function under operation of the dummy function.

Tanaka discloses executing a dummy function in place of a system function when the system function is called (*See col 3, lines 60-65 and col 2, lines 48-55, replacing external or supplemental function for an operating system function*);

executing the system function under operation of the dummy function (*See col 2, lines 59 and col. 6 and lines 32, executing the external or supplemental function or process*); changed execution of the system function and completing execution of the dummy function (*See col 6, lines 8-11, changed the original function func1 execution and completing execution of external or supplemental function func2*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise: executing a dummy function in place of a system function when the system function is called, as taught by Tanaka, in order to expand a programming function without altering the original programming function of the system (*See col 2, lines 58-60*).

**As to claims 47 and 69**, Tanaka also discloses the dummy function by assigning a duplicate calling name to the dummy function and arranging the processor to pre-empt the execution of the system function (*See col 2, lines 53-58, functions having the*

*same names as the replaced function is executed in place of an original system function).*

**As to claims 48 and 70**, Tanaka also discloses the dummy function includes an exit point and an exit program is registered for the exit point (*See col 7, lines 55-65, when original system func1 is called, the original system func1 routine passed control to the external or supplement func1, the external or supplement func1 takes control and adds the new routine or exit program for execution).*

**As to claims 49 and 71**, Tanaka also discloses a method wherein during operation of the dummy function the exit program is executed (*See col 7, lines 60-65 and col 6, lines 8-10, executing the external or supplement routine which added to original system function, whereas that external or supplement routine is a program).*

**As to claims 50 and 72**, Tanaka also discloses a method wherein the execution of the system function is handled by the exit program (*See col 7, lines 55-65, when original system func1 is called, the original system func1 routine passed control to the external or supplement func1, the original system func1 is handled by the external or supplement func1 routine or program).*

**As to claims 51, 52, 73 and 74**, Bills discloses captures and generates copies of the system objects (*See para. [0058], captures the object changes and journals objects).*

Bills does not explicitly disclose the exit program.

Tanaka discloses the exit program (*See col 7, lines 50-65, the original system func1 routine passed control to the external or supplement func1 and executed the external or supplement routine which added to original system function, whereas that external or supplement routine is a program*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise: the exit program, as taught by Tanaka, in order to exit out a programming function when necessary.

**As to claims 53 and 75**, Tanaka also discloses the execution of the system function is handled by the dummy function (*See col 7, lines 55-65, when original system func1 is called, the original system func1 routine passed control to the external or supplement system func1, the original system func1 is handled by the external or supplement system func1 routine*).

**As to claims 54, 55, 76 and 77**, Bills disclose the function captured the system objects and generates copies of the system objects (*See para. [0058], captures the object to see if there are any modification and journals the objects*).

Bills does not explicitly disclose the dummy function and exit program.

Tanaka discloses the dummy function and exit program (*See col 7, lines 50-65, the original system func1 routine passed control to the external or supplement func1 and executed the external or supplement routine, whereas that external or supplement routine is a program*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise: the exit program, as taught by Tanaka, in order to exit out a programming function when necessary.

**As to claims 56 and 78**, Bills disclose the copies of the system objects are saved to disk (*See para. [0038], the objects are stored in file system*).

**As to claims 62 and 84**, Bills also disclose the system objects are one or more of the set of program objects, configuration objects, queues, and space/memory mapped objects (*See para. [0007], objects are programs, files and libraries*).

**As to claims 63 and 85**, Bills also discloses the changed system objects are those system objects that have been created, changed or deleted (*See para. [0005], para [0014] and para. [0058]*).

**As to claims 64 and 86**, Bills inherently discloses wherein the system functions are OS/400 system functions (*See para. [0004], the system functions are operating system functions, and one ordinary skill would recognize that operating system can be OS/400*).

**As to claim 65**, Bills discloses i) executing the system function during which changes to system objects occur (*See para [0058], executes the automatic journaling function when an object modifies or changes* ); and ii) journaling changes to system objects during execution of the system function (*See para [0058], journaling the objects*).



**As to claim 66**, Bills also discloses a method as claimed in claim 65 wherein changes of system objects are journal by integrating journaling commands into the code of the system functions( *See para [0006], the journaling routine is a program included in the operating system modules*).

**Referring to claim 68**, Bills discloses a system for journaling in a database journal hanges to system object including: a processor adapted to generate copies of system objects for journaling (*See para. [0005] and para.[0014], journaling objects changes, e.g. journaling objects when objects are created or deleted in an operating system*; and memory for use by the processor during execution (*See Fig. 1 , system memory*)

Bills does not explicitly disclose executing a dummy function in place of a system function when the system function is called and executing the system function under operation of the dummy function.

Tanaka discloses executing a dummy function in place of a system function when the system function is called (*See col 3, lines 60-65 and col 2, lines 48-55, replacing external or supplemental function for an operating system function*);

executing the system function under operation of the dummy function (*See col 2, lines 59 and col, 6 and lines 32, executing the external or supplemental function or process*); changed execution of the system function and completing execution of the dummy function (*See col 6, lines 8-11, changed the original function func1 execution and completing execution of external or supplemental function func2*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise: executing a dummy function in place of a system function when the system function is called, as taught by Tanaka, in order to expand a programming function without altering the original programming function of the system (See col 2, lines 58-60).

**As to claim 87**, Bills in view of Tanaka inherently discloses wherein the processor is operating under the OS/400 operating system (*See Bills, par 31, lines 1-2, the system functions are operating system functions, and one ordinary skill would recognize that operating system can be OS/400, also see col 5, lines 4-5, processing unit*).

**As to claim 88**, Bills also discloses a computer system for effecting the method of claim 46 (*See Fig 1*).

**As to claim 89**, Bills also discloses a computer system for effecting the method of claim 65 (*See Fig 1*).

**As to claim 90**, Tanaka also discloses a computer readable storage medium tangibly storing software executable by a computer for executing the method of claim 46 (*See col 3, lines 25-26*).

**As to claim 91**, Bills also discloses a computer readable storage medium tangibly storing software executable by a computer for executing the method of claim 65 (*See Fig 1*).

**Claims 57-59 and 79-81 are rejected under 35 U.S.C. 103(a) as being unpatentable by Bills (US PA Pub 2002/0152195 A1) in view of Tanaka (US Patent 6,665,735) and further in view of Owen (US PA Pub 2003/0217031 A1).**

**As to claims 57 and 79,** Bills in view of Tanaka do not explicitly disclose the copies of the system objects are streamed to a database system for journaling.

Owen discloses the copies of the system objects are streamed to a database system for journaling (*See par 34, lines 1-3 and lines 8-14, sending the journal entries to the journal receiver and send to target system, then applies the journal entries to the target system, see also par 35, lines 7-8 and Fig. 5*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise: the copies of the system objects are streamed to a database system for journaling, as taught by Owen, in order to send journal entries from one system to another to assure data integrity of the information system (See par 6, lines 1-3).

**As to claims 58, 59, 80 and 81,** Bills in view of Tanaka do not explicitly disclose the database system is incorporated with a replication system, replicates the copies of the system objects to one or more local or remote databases.

Owen disclose the database system is incorporated with a replication system, replicates the copies of the system objects to one or more local or remote databases (*See par 34, lines 13-18 and Fig. 5, replicating the received journal entries from the journal applying software, the replicated files are reflected in local database*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise the database system is incorporated with a replication system, replicates the copies of the system objects to one or more local or remote databases, as taught by Owen, in order to send journal changes from one system to another to assure data integrity of the information system and to minimize excessive journal data that need to be send (*See par 6, lines 1-6*).

**Claims 60, 61, 82 and 83 are rejected under 35 U.S.C. 103(a) as being unpatentable by Bills (US PA Pub 2002/0152195 A1) in view of Tanaka (US Patent 6,665,735) and further in view of Suzuki (US Patent 6,829, 768 B1).**

**As to claims 60 and 61**, Bills in view of Tanaka do not explicitly disclose wherein messages or exceptions generated by the system function are captured into a queue and the messages or exceptions are forward back to the process by a function.

Suzuki discloses messages or exceptions generated by the system function are captured into a queue and the messages or exceptions are forward back to the original process, (*See col 1, lines 50-60 and col 4, lines 33-37, the adaptor function has a queue manager to captured messages in a queue and forward back to the SDL process, the SDL process is executing operation system tasks*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Bills's system to comprise messages or exceptions generated by the system function are captured into a queue and the

messages or exceptions are forward back to the process by a function, as taught by Suzuki, in order to integrate an external or supplement environment with an original system environment to expand the operations or library of the original system environment and maintain communication between the two environments (*See col 1, lines 20-22 and lines 42-27*).

**As to claims 82 and 83**, they recite essentially the same limitations as claims 60 and 61; therefore, they are rejected based on the same reasons as set forth in claims 60 and 61

**Claim 67 is rejected under 35 U.S.C. 103(a) as being unpatentable by Bills (US PA Pub 2002/0152195 A1) in view of Tanaka (US Patent 6,665,735) and further in view of Cloud (US Patent 6,253,369 B1).**

**As to claim 67**, Bills does not explicitly disclose wherein changes to system objects are journeyed by associating exit points.

Cloud discloses changes to system objects are journaled by associating exit points with the system function and calling an exit program during execution of the system function (*See col 8, lines 27-30 and lines 35-38, allows to custom user exit points for journaling and calling a custom-coded program during executing the system workflow*).

Hence, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify Tanaka's system to comprise journaled by

associating exit points with the system function and calling an exit program during execution of the system function, as taught by Cloud, in order to permit integration of two different system environment with a minimum of integration effort (*See col 3, lines 20-23*).

**(10) Response to Argument**

**A. Rejection of Claims 46-56, 62-66, 68-78 and 84-91 under 35 U.S.C. §103 –**

**Bills in view of Tanaka**

**Independent Claims 46 and 68**

a) Appellant argues Bills does not disclose generating copies of system objects, changed by the execution of the system function, for journaling (See Appellant's Appeal Brief, page 12-14).

**Examiner's Response:**

In response to Appellant's argument, the Examiner respectfully disagrees because the Examiner read "generating copies of system objects" as "generating copies of changes of system objects". In view of Appellant's specification, Appellant indicates the exit program may handle execution of the system function and capture **changes to system objects** occurring during the execution. Copies of changes are generated by the exit program and saved to disk for journaling (See page 3, line 35 and page 4, lines 1-3). Applicant further describes an alternative way to handle execution of the system function and capture changes to system objects during execution by using a dummy

function, and again, Applicant discloses copies of the changes are generated by the dummy function, and the copies of changes to system objects include creation, change and deletion (See page 4, lines 4-8 and lines 21-24). Hence, it is entirely reasonable for Examiner to read "generating copies of system objects" as "generating copies of changes of system objects". Further, the claimed language as it is recited only states "generate copies of system objects, changed by the execution of the system function, for journaling", thus, whether or not Bills is concerned with storing copies of objects after they have changed is unimportant and nearly irrelevant to the claimed invention.

**b) Appellant argues that Tanaka is not analogous art (See Appellant's Appeal Brief, pages 15 and 16).**

**Examiner's Response:**

In response to the Appellant's argument, Examiner respectfully disagrees because the Examiner recognizes that references cannot be arbitrarily combined and there must be some reason why one of ordinary skill in the art would be motivated to make the proposed combination of references. *In re Nomiya, 184 USPQ 607 (CCPA 1975)*. However, the test for combine references is what the combination taken, as a whole would suggest to one of ordinary skill in the art. *In re McLaughlin, 170 USPQ 209 (CCPA 1971)*. References are evaluated by what they suggest to one versed in the art, rather than by their specific disclosures. *In re Bozek, 163 USPQ 545 (CCPA 1969)*. In this case, the Examiner relied on Tanaka to teach the programming method, e.g. Tanaka discloses replaces [changes] execution from a normal or original function to a

dummy function to perform other operations, discontinues the execution of the dummy function when the other operations are done and resume the execution of the normal or original function (See col 2, lines 45-60 and col 6, lines 8-32). The system described by Tanaka is a system written in C programming language with a C compiler (See col 1, lines 55-60). This system has a programming method ready to be implemented or operated in any environments that are well known in the art, for example, environments such as journaling, backup or replication or any other environments. A hardware system alone cannot be functional unless a software program gives instruction to that hardware system; in fact, the journaling system of Bills comprises programs include routines, objects, components and data structures (See para. [0030]). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the Appellant's invention to modify the journaling system of Bills to utilize a dummy function in place of a system function when the system function is called, as taught by Tanaka, the motivation being to have enhanced the system of Bills by expanding a programming function without altering the original programming function of the system (See col 2, lines 58-60).



**Dependent Claims 47 and 69**

**Appellant argues Tanaka cannot pre-empt the execution of the system function when a dummy function is called because Tanaka is concerned with design-time functionality, not run-time functionality (See Appellant's Appeal Brief, page 16).**

**Examiner's Response:**

The Examiner's reasoning found in responses to claims 46 and 68 under this Examiner's Answer, herein is applicable to, and addresses this argument. Further, the claimed language as it is recited is silence on executing the system function in run-time functionality, thus whether or not the Tanaka's system is implemented at a lower level functionality or design-time functionality is irrelevant and unimportant to the claimed language. Examiner further notes that Tanaka's programming method can be implemented in a "higher level" or "run-time level" for the sake of user or system preferences, and even though the "higher level" or "run-time level" implementation, if its any different from the lower level or design level, the system objects are necessarily somehow passed on to a lower level or design level implementation to allow proper execution, and lastly, in either case, Tanaka sets out and achieves the same result as the claimed invention.

**Dependent Claims 53 and 75**

**Appellant argues Tanaka cannot pre-empt the execution of the system function when a dummy function is called because Tanaka is concerned with design-time functionality, not run-time functionality (See Appellant's Appeal Brief, page 16).**

**Examiner's Response:**

The Examiner's reasoning found in responses to claims 46 and 68 under this Examiner's Answer, herein is applicable to, and addresses this argument. Further, the claimed language as it is recited is silent on executing in run-time functionality, thus whether or not the Tanaka's system is implemented at a lower function or design-time functionality is irrelevant and unimportant to the claimed language. Examiner further notes that Tanaka's programming method can be implemented in a "higher level" or "run-time level" for the sake of user or system preferences, and even though the "higher level" or "run-time level" implementation, if its any different from the lower level or design level, the system objects are necessarily somehow passed on to a lower level or design level implementation to allow proper execution, and lastly, in either case, Tanaka sets out and achieves the same result as the claimed invention.

For the above reasons, it is believed that the rejections should be sustained.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer

An appeal conference was held on 19 April 2010 with below listed conferees.

Respectfully submitted,

Conferees:

/Yuk Choi/  
Examiner. Art Unit 2164

Charles Rones  
/Charles Rones/  
Supervisory Patent Examiner, Art Unit 2164

James Trujillo  
/James Trujillo/  
Supervisory Patent Examiner, Art Unit 2159

PTG:jr

901 North Glebe Road

Arlington, VA 22203-1808

(703) 816-4100